Object-Oriented Programming

04/16/2020

Lab Assignment 5

1. Create a class `Person` that stores person's information: name and age. Then create another class `PersonVector` that stores many persons in its data member which uses a `vector`. Please place the declarations of these two classes in separate header (.h) files with header guards, and implement the classes in the corresponding (.cpp) files.

   For the `Person` class, please provide a two-argument constructor, and getter methods for the name and age.

   In this problem, please make data members all private and member functions all public. The following is the main program you cannot change:

```
#include "PersonVector.h"
#include <iostream>

int main(){
    PersonVector pv;
    pv.addPersons();
    pv.displayPersons();
}
```

   The following are sample runs:

```
How many person(s) to add?2
1 Person to add:John 18
2 Person to add:Jane 21
John    18
Jane    21
```

```
How many person(s) to add?4
1 Person to add:Adam 1
2 Person to add:Clark 14
3 Person to add:Claire 18
4 Person to add:Louis 42
Adam    1
Clark   14
Claire  18
Louis   42
```

2. Create two classes `Rectangle` and `Square`. Please place the declarations of these two classes in separate header (.h) files with header guards, and implement the classes in the corresponding (.cpp) files.

If one of your classes depends on the other class before the other class can be defined, you can forward declare it. For example, if you need to know the existence of the `Rectangle` class in the `Square` class, you can forward declare before the `Square` class in Square.h:

```
class Rectangle;

class Square{
    // actual declaration of Square
    // ...
};
```

The `Square` class has a private data member side, and a public single argument constructor.

The `Rectangle` class has two private data members width and height, and three public member functions `area()`, `convert(Square s)`, and `print()`.

You can make these two classes `friend`s. But only do so for the necessary part. The following is the main method you cannot change:

```
int main () {
  cout << "Please input the side of the square: ";
  int side;
  cin >> side;

  Rectangle rect;
  Square sqr(side);
  rect.convert(sqr);

  cout << "The converted Rectangle has:" << endl;
  rect.print();

  return 0;
}
```

The following are sample runs:

```
Please input the side of the square: 3
The converted Rectangle has:
width: 3, height: 3, and area: 9
```

```
Please input the side of the square: 5
The converted Rectangle has:
width: 5, height: 5, and area: 25
```